A Quantitative Comparison of Ad Hoc Routing Protocols with and without Channel Adaptation

Xiao-Hui Lin, Yu-Kwong Kwok, *Senior Member*, *IEEE*, and Vincent K.N. Lau, *Senior Member*, *IEEE*

Abstract—To efficiently support tetherless applications in ad hoc wireless mobile computing networks, a judicious ad hoc routing protocol is needed. Much research has been done on designing ad hoc routing protocols and some well-known protocols are also being implemented in practical situations. However, one major imperfection in existing protocols is that the time-varying nature of the wireless channels among the mobile terminals is ignored, let alone exploited. This could be a severe design drawback because the varying channel quality can lead to very poor overall route quality in turn, resulting in low data throughput. Indeed, better performance could be achieved if a routing protocol dynamically changes the routes according to the channel conditions. In this paper, we first propose two channel adaptive routing protocols which work by using an adaptive channel coding and modulation scheme that allows a mobile terminal to dynamically adjust the data throughput via changing the amount of error protection incorporated. We then present a qualitative and quantitative comparison of the two classes of ad hoc routing protocols. Extensive simulation results indicate that channel adaptive ad hoc routing protocols are more efficient in that shorter delays and higher rates are achieved, at the expense of a higher overhead in route set-up and maintenance.

Index Terms—Mobile computing, ad hoc networks, routing protocols, on-demand routing, channel adaptation.

✦

1 INTRODUCTION

To realize efficient information exchange in a peer-to-peer manner (i.e., without infrastructure support) in an ad hoc wireless mobile computing network [6], [10], [12], [21], [29], [33] (e.g., a network consisting of personal digital assistants, notebook computers, and cell phones is formed in an ad hoc manner to perform file swapping), a judicious routing protocol is needed for the source to locate the destination in the network [5], [26] and set up a path to reach it. There are two major classes of ad hoc routing protocols: on-demand and table based. As many researchers have pointed out [2], [16], table-based algorithms are notoriously inefficient in that they require periodic update of the routing information stored in the routing tables, even when there is no data traffic. The major merit of table-based algorithms as compared with on-demand algorithms, is that the set up delay for a data transfer is expected to be shorter because a route is presumably stored in the table for use. However, such a route may no longer exist or be usable when the actual data transfer is to take place for at least two reasons. First, due to the mobility of the mobile terminals in the network, their geographical locations may have changed when a data transfer is required, rendering a previously obtained route useless. The second reason, which, we believe, is a more important one, is that the quality of the

channels among the mobile terminals is inevitably time-varying (due to signal propagation effects such as shadowing and fast fading [25]) and, thus, the links in a route may no longer be usable even if the geographical locations do not change much. Indeed, this is a major consideration overlooked in previous researches on ad hoc routing protocols [2], [4], [16], [26], [27], [28], [32]. The challenge is to design ad hoc routing protocols that can perform well in the following two aspects:

- *Connectivity*: The protocol can swiftly identify feasible routes for connecting peer ad hoc network hosts and
- *Link Quality*: The protocol is able to select links that have high bandwidth to form the routes and, more importantly, to rebuild the routes when the channel quality of some links has already deteriorated.

Recently, there have been several attempts to design ad hoc routing protocols that meet the above challenge [7], [14], [17], [18], [19]. In our study, we first propose two new channel adaptive routing protocols, called BGCA (Bandwidth Guarded Channel Adaptive) and RICA (Receiver-Initiated Channel Adaptive) protocols, and then perform a quantitative comparison of protocols with and without exploiting channel adaptation.

This paper is organized as follows: In the next section, we describe, in detail, the five routing algorithms considered in our study. For an excellent survey on ad hoc routing protocols, the reader is referred to [28]. Section 3 contains the performance results and interpretations of a quantitative comparison of the five algorithms. We provide some concluding remarks in Section 4.

2 AD HOC ROUTING PROTOCOLS

In this section, we describe, in detail, the five protocols considered in our performance study. For the two

- *X.-H. Lin is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Room 807, Chow Yei Ching Building, Pokfulam, Hong Kong. E-mail: xhlin@eee.hku.hk.*
- *Y.-K. Kwok is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Room 604, Chow Yei Ching Building, Pokfulam, Hong Kong. E-mail: ykwok@eee.hku.hk.*
- *V.K.N. Lau is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Room 516, Chow Yei Ching Building, Pokfulam, Hong Kong. E-mail: knlau@eee.hku.hk*

channel-adaptive protocols proposed by us, we also provide some illustrative examples to further highlight their distinctive properties.

2.1 The Associativity-Based Routing (ABR) Protocol

ABR [16], [28], [31], [32] is a source-initiated on-demand routing protocol—a mobile terminal in the network does not need to keep a route to every other terminal. The major distinctive characteristic of ABR is that the route is not chosen on a shortest-path basis as in other protocols, but on a long-lived basis. Selecting a long-lived route has much merit, such as the chosen route is more robust (not easy to break due to mobility), the maintenance of route is easier and the number of route reconstruction (RRC) is reduced (thus, the routing overhead is reduced and more bandwidth is saved). In ABR, longevity of a route is considered as the top priority. In other protocols, such as Link State and AODV, the primary goal is to find the shortest path. Consequently, in these latter protocols, data flow transmission interruptions occur more often and more route reconstructions are needed. The essence of ABR is that, as a mobile terminal moves, its associativity with the neighbor terminals also changes and this associativity can be quantified by using associativity ticks. In ABR, each mobile terminal periodically transmits beacons (such as hello messages) to signify its existence. When these beacons are received by its neighbor terminals, the beacons cause the associativity of this terminal with its neighbor terminals increase. The greater the associativity is, the more stable this terminal will be. A high associativity of the terminal means a low mobility of a terminal. If a terminal A moves out of the transmission range of another terminal, the associativity record of A in this terminal is reset. These mechanisms are designed to achieve the most fundamental objective of ABR—to find a long-lived route between the source and destination terminals.

2.2 The Ad Hoc On-Demand Distance Vector (AODV)

AODV [26], [27], [28] is a purely reactive routing protocol. In this protocol, each terminal does not need to keep a view of the whole network or a route to every other terminal. Nor does it need to periodically exchange route information with the neighbor terminals. Furthermore, only when a mobile terminal has packets to send to a destination does it need to discover and maintain a route to that destination terminal. The major distinctive characteristics are:

- Routes are discovered or maintained only when necessary (i.e., on-demand).
- AODV is loop-free [8], [27] at all times and this is accomplished through the use of a terminal sequence number which is incremented monotonically (this technique also ensures that the most current route is always used when discovering a route).
- AODV is a bandwidth efficient routing algorithm, which greatly reduces the use of limited bandwidth (the number of route broadcasts are minimized on as-needed basis).

- AODV responds very quickly to the topology changes of the network and can recover a broken route in a timely manner.
- AODV has a low storage requirement on the terminal because it only needs to maintain the active neighbors information instead of a full route to the destination.

In AODV, each terminal contains a route table for a destination. A route table stores the following information: destination address and its sequence number, active neighbors for the route, hop count to the destination, and expiration time for the table. The expiration time is updated each time the route is used. If this route has not been used for a specified period of time, it is discarded.

2.3 Link State Routing Protocol

To have a more comprehensive comparison, we also consider the classical Link State routing algorithm which is a proactive approach. The link state routing protocol was originally designed for the fixed-line network [11]. In this protocol, each mobile terminal keeps its own view of the whole network. When a mobile terminal has a packet to forward, it uses a shortest path algorithm (in our study, the Dijkstra algorithm [3] is used) to determine the next hop to forward that packet to its destination. Every mobile terminal must keep an up-to-date view of the network. When a mobile terminal finds a link cost change (due to the change in channel quality) with one of its neighbors, it floods this change throughout the network. Once this flooding packet is received by a mobile terminal, this mobile terminal updates its view of the network accordingly. Here, it should be noted that asynchronous updates can cause transient routing loops, but such a loop is eliminated eventually as the flooding packet is propagated throughout the network. In our study, the flooding packet is propagated through the common broadcast channel shared by all mobile terminals (detailed in Section 3).

2.4 Bandwidth-Guarded Channel Adaptive (BGCA) Routing

The main idea of BGCA is that, when a link is in deep fading, the upstream terminal executes a local search to find a partial route to the destination [17]. It is designed based on AODV, but also incorporates some ingredients of ABR, e.g., the local search for a partial route and selection of the shortest route. Before describing the protocol in detail, we introduce the channel model we used in our study below. Specifically, throughout the paper, we assume that a multicode TDD (time-division duplex) CDMA MAC (multiple access control) layer [13] is used in all the protocols.

2.4.1 Channel Model

The wireless channel between every two mobile terminals is time-varying and is governed by physical effects in signal propagation: path loss, shadowing, and microscopic fading [25]. Path loss refers to the change in received signal strength versus the distance between the transmitter and the receiver. Shadowing loss refers to the change in received signal strength due to variations in terrain structure. These two factors fluctuate in macroscopic time scale (5-10 seconds). On the other hand, microscopic fading refers to the
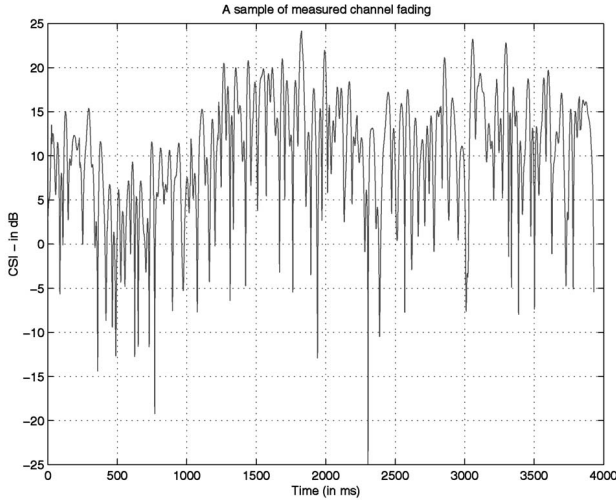
Fig. 1. A sample of the fluctuating quality of a wireless channel.

variation of signal strength due to multipath propagation and, as such, fluctuates in microscopic time scale (inversely proportional to user mobility). In this paper, we target various levels of mobility (average speed varies between 0-75 km/hr) and, therefore, the coherence time of the fading channel is of the order of 80-100 ms. The combined effect (as shown in Fig. 1) of the above signal propagation factors is characterized by the CSI (Channel State Information), which is the measured SNR (signal-to-noise ratio) of known pilot symbols on a feedback channel. Since a packet or physical frame duration in our system is 1 ms,[1] it is justified for us to assume that the CSI remains approximately constant for the duration of at least one frame.

In this paper, we adopt the adaptive physical layer design called ABICM[2] [15], in which variable throughput modulator and channel coding are used (see Fig. 2). Specifically, when CSI is available at the transmitter, the transmitter performs "burst-by-burst" throughput adaptation [15] with respect to the CSI. For instance, when the CSI indicates that the channel is of a good quality, the transmitter employs a high-order modulation (e.g., 16QAM) and high-rate error correction code so as to boost the instantaneous throughput. On the other hand, when the channel quality is poor, the transmitter employs a lower-order modulation (e.g., BPSK) and low-rate error protection so as to protect the packet transmission at the expense of lower instantaneous throughput. Fig. 3 illustrates the instantaneous throughput variation with respect to the CSI. Indeed, the salient concept of adaptive physical layer has been widely deployed in various wireless systems such as 3G1x, EV-DV, UMTS, HSDPA, and IEEE 802.11a/g [24].

In our study, we use a 4-mode ABICM configuration [15] and, thus, there are four distinct possible throughput levels. To illustrate, let us consider a simple example. Based on the CSI of the channel, we divide the channel quality into four classes: A, B, C, and D, with a throughput of 250 kbps, 150 kbps, 75 kbps, and 50 kbps, respectively (after adaptive

channel coding and modulation). We define a CSI-based "hop" in the following manner: If a link between two terminals with a channel quality of class A (with throughput of 250 kbps), then the distance between two terminal is defined as *one hop*. We use this "distance" as a baseline metric. Then, if a link between the two terminals with a channel quality of class B (with a throughput of 150 kbps), the distance between the two terminals is considered as 1.67 hops because the transmission delay now is 1.67 times compared with a link of class A. Thus, a link has a throughput of 250 kbps, 150 kbps, 75 kbps, and 50 kbps, the distances are 1, 1.67, 3.33, and 5 hops, respectively.

2.4.2 Path Discovery

When a terminal has packets to send to a destination terminal, it first generates an RREQ (route request) packet, which includes the following fields: type of the packet, source and destination IDs, broadcast ID, bandwidth requirement (e.g., if the source generates 10 packets per second with a packet size of 1 kbits, then the bandwidth requirement is 10 kbps), hop count from the source (CSI based), and intermediate terminals list. Every time the source broadcasts an RREQ to the same destination, the broadcast ID increases by one. The source terminal appends its ID to the intermediate terminals list and broadcasts this RREQ throughout the network in search of the destination terminal.

When the RREQ is received by an intermediate terminal, the intermediate terminal checks its history hash table to see whether it has processed this packet before. If so, it discards the packet. If it has not seen this packet before, it checks whether the available bandwidth of the link between it and the terminal from which the RREQ comes can satisfy the bandwidth requirement of the RREQ (also take into consideration the used bandwidth by other sessions). If the available link bandwidth cannot satisfy the requirement, the terminal also discards the RREQ; otherwise, the intermediate terminal records this RREQ in its history hash table the following information: source and destination IDs and broadcast ID. Then, the intermediate terminal appends its ID to the intermediate terminals list of the RREQ and sets the hop count field of RREQ to the original hop count plus the hop distance from the upstream terminal. After doing all these steps, this intermediate terminal rebroadcasts the RREQ out. This process continues until the RREQ reaches the destination terminal. Eventually, the destination terminal may receive several RREQs from all possible routes.

Fig. 4a shows the broadcast of RREQ in the network. Each RREQ includes a full route from the source to the destination and related hop distance (CSI-based). The destination terminal chooses the shortest route and sends a route reply (RREP) along this route to the source as shown in Fig. 4b (three routes with a hop distance of 9.33, 10.33, 11, respectively). The RREP includes the full list of terminals along the route so that each terminal knows its upstream and downstream terminals together with the related CDMA PN (pseudorandom noise) codes[3] to receive and transmit packets.

---

1. As a practical reference, the longest frame duration in the IEEE 802.11a standard is approximately 800 microseconds [24].
2. Using the ABICM scheme is just for illustration only, other schemes (e.g., the onse suggested in [1]) can also be used. For details of the ABICM scheme and its applications in MAC protocols, the reader is referred to [15].

3. We assume CDMA (code division multiple access) is used as the underlying MAC (multiple access control) layer.
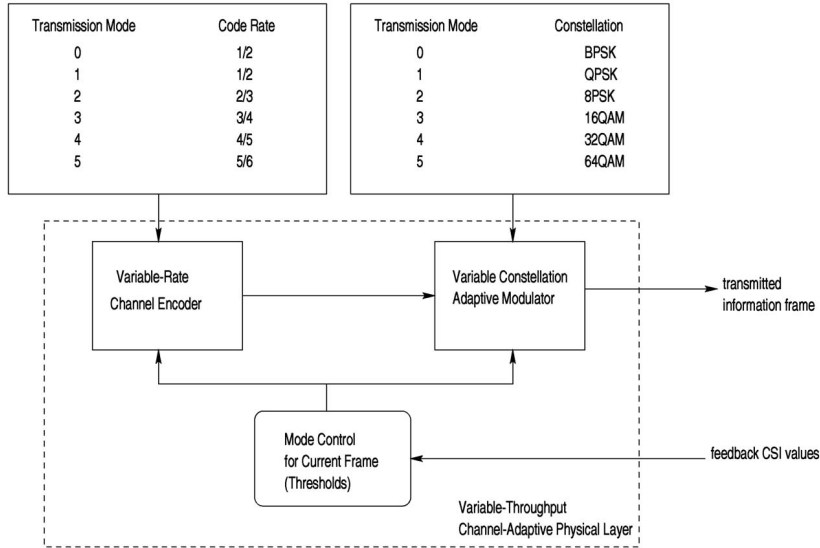
Transmission Mode	Code Rate
0	1/2
1	1/2
2	2/3
3	3/4
4	4/5
5	5/6

Transmission Mode	Constellation
0	BPSK
1	QPSK
2	8PSK
3	16QAM
4	32QAM
5	64QAM

Fig. 2. A conceptual block diagram of a variable throughput adaptive channel modulator and channel coding system.

2.4.3 Route Maintenance

A long period of deep channel fading leads to the drop of data throughput, network congestion, and, consequently, data loss. In BGCA, each terminal can adjust its data transmission rate according to the channel quality by using the adaptive channel coding and modulation scheme ABICM [15] as mentioned earlier. Specifically, when a downstream terminal notices channel quality degradation, it informs the upstream terminal of this CSI change through the feedback channel. The upstream terminal checks whether the throughput of this link can still satisfy the bandwidth requirement of the data traffic. In our simulations, we stipulate that the bandwidth of the link must be greater than or equal to 75 kbps (i.e., at least class C) and must be greater than or equal to the traffic loads. If the link can still satisfy the requirement, the upstream terminal ignores this warning. If the link deteriorates too much and cannot satisfy the bandwidth requirement, the upstream terminal has to perform local search to find a partial route to the destination. If several routes share this degraded link,

then the upstream terminal decides which routes have to perform local search using the method described below.

The strategy is to let the route whose partial route is shorter (with smaller hop count) to perform this local search (only if the link is used by several routes, but, normally, this is not the case in the simulations, as detailed in Section 3), so as to shorten the search time. The upstream terminal of the link in fading first generates an LQ (local query) packet which includes the following information: type of the packet, source and destination IDs, ORIG ID (the ID of the terminal originating the LQ), sequence number, intermediate terminals list, bandwidth requirement, hop count (originally set to zero), and TTL (time-to-live) field. The TTL field is set to the hop distance (not CSI-based) from the originating terminal to the destination; each time the LQ is rebroadcast, the TTL is decreased by one. When the TTL reaches zero, the LQ is discarded. In this manner, a full broadcast is avoided.

When this LQ is received by a neighbor terminal, this terminal first checks whether the packet comes from a link that can satisfy the bandwidth requirement in the LQ; if not, this LQ is discarded. If the bandwidth requirement is satisfied, the terminal further checks whether it has seen this packet before. If it does, the LQ is also discarded; otherwise, the neighbor terminal appends its ID to the intermediate terminals list in the LQ, resets the hop count (CSI based), and rebroadcasts the LQ out. This process continues until the LQ reaches the destination. At last, the destination may receive several LQs from the originating terminal through all possible routes. Each LQ includes a full route from the originating terminal to the destination and the hop distance of the route. Now, the destination terminal can select the shortest partial path and send an RREP including all the terminals in the path to the originating terminal.

On receiving the RREP, the originating terminal uses this route to substitute for the original partial route. If several intermediate terminals broadcast the LQ at the same time, the partial route is eventually determined by the most
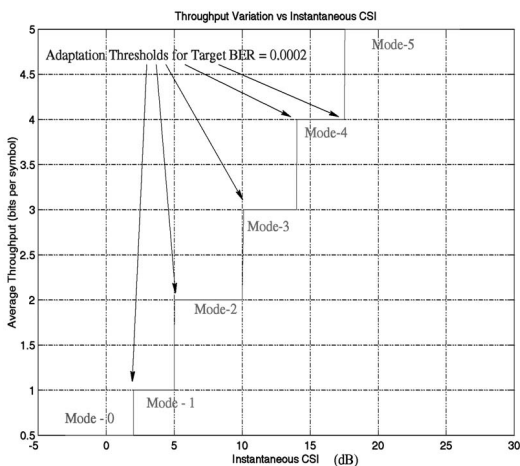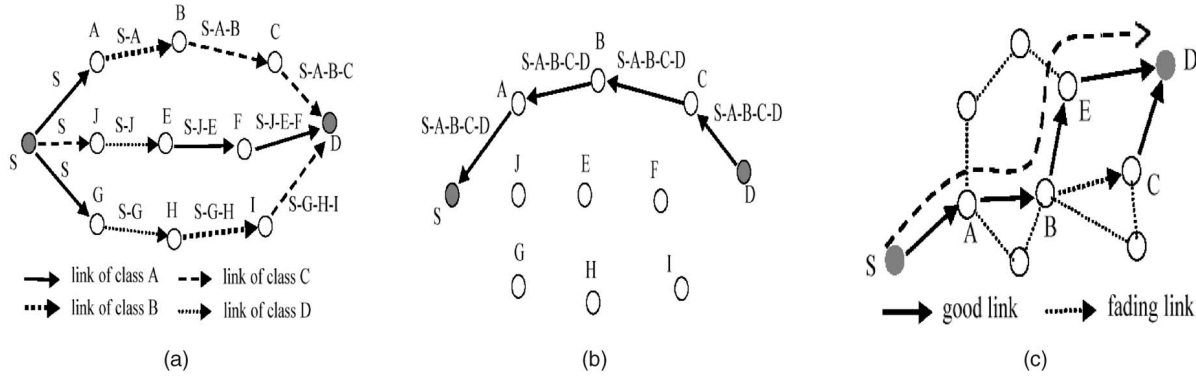
Fig. 3. Average throughout versus instantaneous CSI.

Fig. 4. Illustration of the BGCA protocol. (a) Broadcast of RREQ, (b) generation of RREP, and (c) change of links.

upstream terminal. If the timeout period has passed and the originating terminal has not yet received the RREP, it performs another trial. If it still fails, it unicasts an unreachable message to the source. On receiving this unreachable message, the source initiates another RREQ in search of the destination. In Fig. 4c, the original route from source to destination is S-A-B-C-D. Link B-C is in deep fading and cannot satisfy the bandwidth requirement, so terminal B performs the local search. Now, the new full route is S-A-B-E-D.

2.4.4 Link Breakage

When an intermediate terminal or the destination moves and causes the breakage of a link, and the upstream terminal notices that (because it cannot receive the ACKs of the data packets), it unicasts a route error (RERR) to the source terminal. On receiving this RERR, the source initiates an RREQ in search of a new route to the destination. If the source terminal moves and causes the breakage of the link, it initiates an RREQ to find a new route to the destination.

2.5 Receiver Initiated Channel Adaptive (RICA) Routing

RICA is also a channel adaptive routing algorithm. The major feature of RICA is to make use of this channel-varying property and let the routing between source and destination adapt to the CSI of the whole route and we call this adaptive-routing-based on CSI. This means that, in RICA, the entire route between the source and destination terminals is changing with time, instead of just changing a few links, as in the BGCA algorithm. Thus, while BGCA performs localized route refinement in response to link quality changes, RICA performs globalized route reconstruction.

2.5.1 Route Discovery

In the RICA protocol, the source does not keep a route to any possible destination unless it is necessary (when it has packets to send to that destination). When the source terminal has packets to transmit, it must first find a route to the destination. The source terminal generates an RREQ packet which includes the following information: type of the packet, source address, destination address, hop-count from the source, and broadcast ID of RREQ. Each time the source generates an RREQ, the broadcast ID increases by

one. The source and destination addresses together with the broadcast ID uniquely identify an RREQ.

After the source generates this RREQ, it sets the hop-count field to zero and broadcast this packet out in search of the destination terminal. Any intermediate terminal receiving this RREQ first checks whether it has seen this packet before by looking up its history table. If so, this packet is discarded; otherwise, the terminal records this packet in its history hash table including the following information: source and destination addresses and broadcast ID. The intermediate terminal must also remember its upstream terminal from which it receives the first RREQ so that it knows to which terminal it should forward the RREP (if this terminal is in the route chosen by the destination). The intermediate terminal also measures the CSI of the link through which this RREQ comes and computes the related hop distance from the upstream terminal. Then, the intermediate terminal resets the hop-count to the original value plus the hop distance to the upstream terminal. After doing all these steps, the terminal rebroadcasts this RREQ packet to its surrounding terminals. This process continues until the RREQ reaches the destination.

Fig. 5a shows the broadcast of RREQ in the network. In the end, the destination terminal receives several RREQs of the same source from all possible routes. The destination also knows the hop-count (as defined earlier) distance of these routes and it chooses a route with the minimal hop-count. In Fig. 5a, an RREQ reaches the destination terminal through three routes (note that the links are labeled with the channel classes) with the hop count distances 6, 7, and 4.33, respectively. The destination terminal generates an RREP which includes the following information: type of the packet, source address, destination address, sequence number (corresponding to the RREQ), and hop count of the route. The destination terminal then unicasts this RREP along the route (shortest in hop count) to the source terminal (note that each terminal knows its upstream terminal to which to forward the RREP), as shown in Fig. 5b. After receiving this RREP, the intermediate terminal sets its route entry to the destination as valid and updates the following information: the source and destination addresses and the next upstream and downstream terminals (it also begins to use the related PN codes to receive
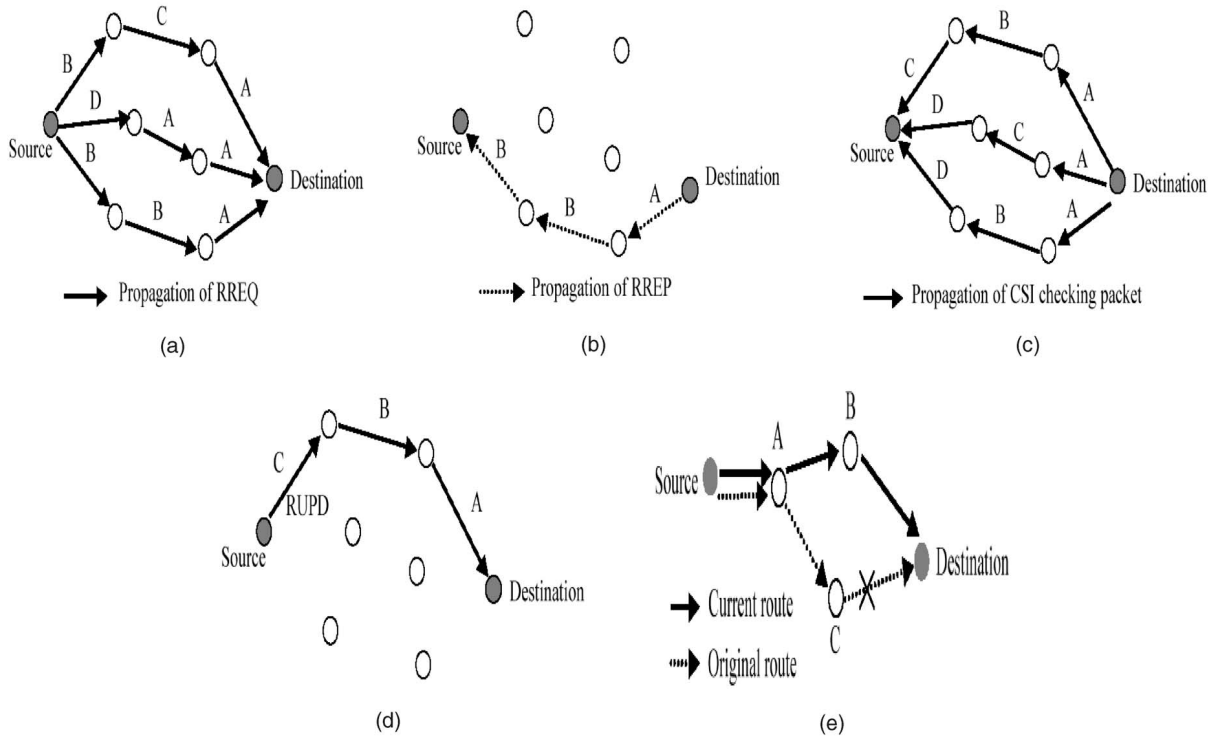
Fig. 5. Illustration of the RICA protocol. (a) Broadcast of RREQ, (b) generation of RREP, (c) receiver initiated CSI checking, (d) generation of RUPD by the source, and (e) change of route.

and transmit packets). When the RREP reaches the source, the source can transmit packets to the destination terminal.

2.5.2 Broadcast of CSI-Checking Packets

Because the channel quality between two terminals is a time-varying function, the throughput of the route to the destination is also changing with time, as discussed earlier. The essence of RICA is to make routing to the destination adaptive to this signal fading environment (due to multipath propagation) such that, ideally, a route can be identified with high throughput to the destination all the time, even in this channel quality fluctuating condition. The idea is to let the destination terminal broadcast a CSI checking packet periodically (for example, three to four seconds, this has to be determined according to the degree of CSI fluctuation, which we call the "channel change speed"). This checking packet is used to measure the CSI of every link it traverses, then converts this CSI value to hop-count. Eventually, the source receives several CSI checking packets from the destination, then it can choose the shortest one as the new route.

We explain this process in more detail with the help of Figs. 5c and 5d. First, the destination terminal generates a CSI checking packet which includes the following information: type of the packet, source and destination IDs, broadcast ID, hop-count[4] (based on CSI) from the destination, and TTL (time-to-live) field. Every time the destination broadcasts a new CSI-checking packet, the broadcast ID increases by one. TTL is used to limit the broadcast scope of the packet because a full broadcast is avoided to save

4. The hop count is set to zero at the beginning.

bandwidth. The TTL field is set to the originally known hop distance (not based on CSI) of the path.

Every time the packet is rebroadcast, the TTL field is decreased by one and, when the TTL is zero, this packet is discarded. An intermediate terminal receiving this checking packet resets the hop-count field based on CSI as mentioned before and decreases the TTL by one and rebroadcast this checking packet out. In this rebroadcast checking packet, the intermediate terminal must specify from which terminal (e.g., this terminal is possibly a downstream terminal) it receives this checking packet.

In the above-mentioned manner, a downstream terminal can also overhear this checking packet, and then sets the intermediate terminal as its potential upstream terminal and knows with which CDMA PN code the upstream terminal uses to send packets to it. It sets its route entry to the destination terminal as valid. Now, it is ready to receive packets from the possible upstream terminal and continues detecting this PN code for a period of time (if this PN code now is not used to receive packets, we set this detecting period to 100 ms; during this period, if no packet to the destination is transmitted using this PN code, the terminal stops detecting the PN code and sets the route entry to the destination as invalid). The intermediate terminal must also remember the downstream terminal from which it receives the first checking packet (in the future, it uses the corresponding PN code to send packets to this downstream terminal).

Note that a terminal only broadcasts a checking packet once and if it receives further copy of the same packet, it discards the duplicate. In the end, the source terminal receives several checking packets from all possible routes as

shown in Fig. 5c. The source terminal then selects the shortest path and uses it to substitute for the original route. Three candidate routes in Fig. 5c with hop-counts of 6, 9.33, and 7.67, respectively. The source terminal then sends a route update (RUPD) packet to its next downstream terminal as shown in Fig. 5d. Subsequently, this downstream terminal is also ready to receive packets and set the route entry as valid. Up to this point, the new route is set up and can be used to transmit packets. The first transmitted packet has an update field. On receiving this first packet, the downstream terminal updates the route entry including the following fields: source and destination IDs, upstream and downstream terminals, and sets the downstream terminal from which it receives the first CSI checking packet as the next hop to the destination. Note that:

- The original route will at last automatically expire (after an idle period, for example, 3 seconds) and be deleted.
- The breakage of the link in a previous route has no effect on the data transmission in the current route if this link is not in the current route, this is illustrated in detail below.

2.5.3 Route Maintenance

In RICA, the update of the route entry can be very frequent and, thus, an upstream terminal A must be sensitive to the connection with its downstream terminal B. To ensure that, the downstream terminal B sends an ACK (acknowledgment) packet to confirm the receipt of data packet. This ACK packet is sent with another PN code (note that A sends packet to B using the PN code PN(A, B), while B sends packet to A using PN code PN(B, A); these two codes are different). When a terminal notices that its downstream terminal has moved out of its transmission range, it generates an RERR (route error) packet which includes the following information: source and destination IDs and the terminal's own ID. The terminal then unicasts RERR to the upstream terminal. The upstream terminal first checks whether the terminal unicasting the RERR is its downstream terminal by looking up the related route entry. If not, it ignores this RERR because this RERR comes from a broken route which is out of date and has no effect on the data transmission that is going on in the current route.

For example, in Fig. 5e, terminal C finds that the link to destination is broken and sends an RERR to A, but A ignores this RERR because it knows that terminal C is not its downstream terminal and the RERR comes from an old link that is not used by the current route. If the terminal unicasting the RERR is its downstream terminal, it also unicasts this RERR to its upstream terminal. The process continues and, if the RERR reaches the source, the source can decide whether it should initiate another RREQ based on two considerations: 1) The source terminal now is receiving CSI checking packets and, in this case, the source terminal ignores the RERR and chooses the shortest route based on CSI checking packets and 2) if the source terminal is not receiving CSI checking packets, it broadcasts an RREQ in search of the destination and waits for an RREP. There are three possible scenarios:

- If the RREP reaches the source together with the CSI checking packets (the source terminal waits 40 ms so that it may receive all the CSI checking packets; during this period, RREP also reaches the source), the source selects the shortest route based on CSI checking packets and RREP packet.
- If the CSI checking packets arrive before the RREP, the source decides the route based on these CSI checking packets; afterward, if RREP also arrives, the source chooses the route based on RREP.
- If the RREP arrives before the CSI checking packet, the source chooses route based on RREP; afterward, the CSI checking packets arrive, the route is decided based on CSI checking packets.

3 RESULTS AND COMPARISON

In this section, we present the results obtained in our extensive simulations using our enhanced NS-2 platform [23] comparing the five algorithms considered in this paper. We first introduce the simulation environment.

3.1 Simulation Environment

The simulation parameters we used are as follows:

- number of terminals: 50,
- testing field: $1,000m \times 1,000m$,
- mobile speed: uniformly distributed between 0 and MAXSPEED (will be elaborated later),
- mobility model: the *modified* random way-point model [34]—when the terminal reaches its destination, it pauses for seconds, then randomly chooses another destination point,
- radio transmission range: 250 m,
- channel model: using a 4-mode ABICM [15] (adaptive modulator and coder), with throughputs of 250 kbps, 150 kbps, kbps, and 50 kbps, respectively,
- bandwidth of the common channel: 50 kbps (this channel is robust in that it uses the most conservative error protection level and, thus, can withstand deep fading and interference),
- MAC of common channel: unslotted CSMA/CA based on CDMA [13],
- traffic load: 10 terminal pairs and in each pair, we change the traffic load for 10, 15, 20, and 60 packets/ sec, respectively.

Furthermore, the size of the data packet is 512 bytes and the capacity of data buffer size is set to be 10 packets for each connection of two adjacent mobile terminals when traffic load is low (i.e., at 10, 15, and 20 packets/sec) and 50 packets when the load is high (i.e., at 60 packets/sec), respectively. Doing so, we can have a fair comparison of the protocols under a different load and we can avoid having the buffer size to become the major bottleneck of the protocol's performance when the load is extremely high. However, buffer size is still a limiting factor. Specifically, when the link is in deep fading, and the packets cannot be sent out timely, congestion results, this leads to dropping of the packets. Our aim is to test the robustness of all the algorithms under a modest but reasonable allocation of buffer given the same channel fading wireless environment.

TABLE 1
Summary of Protocol Characteristics

	ABR	AODV	Link State	BGCA	RICA
Power Consumption	Low	Low	High	Low	Medium– High
Loop Free?	Yes	Yes	No	Yes	Yes
Routing Metric	Associativity of terminal and link load	Shortest path	Shortest path (CSI-based)	Shortest path (CSI-based)	Shortest path (CSI-based)
Alternative Route Available	No	No	Yes	No	No
Adaptive to CSI	No	No	Yes	Yes	Yes
Protocol Type	Demand- Driven	Demand- Driven	Table-Driven	Demand- Driven	Demand- Driven

The transmission of packet is a store-and-forward process. When packet reaches an intermediate terminal, it waits in the buffer queue for service (FCFS). Each packet is allowed to be kept in the buffer for no more than three seconds and if it has not been transmitted in this period, it is discarded. The generation of data packets in each source terminal follows a Poisson arrival process, i.e., the inter arrival of two packets is exponentially distributed.

Each simulation is run for 500 seconds (simulation time) and repeated for 25 trials. We compute the average of the results of these 25 sets of data. To evaluate the five routing algorithms, we compare them using three major metrics:

- **Average End-to-End Delay**: Measured in ms, the end-to-end delay includes the processing time and queuing time of packet in each terminal in the route.
- **Successful Percentage of Packet Delivery**: This is the ratio of packets reaching the destination to total packets generated in the sources. A packet may be dropped if there is not enough data buffer due to the congestion or has stayed in the buffer for more than three seconds.
- **Routing Overheads**: This is measured in bps. We count the total amount of routing packets and data acknowledgment packets in each round of simulation. Each time the common channel is used to transmit a routing packet, this is counted as one transmission. We average the amount of routing overheads (in bits) to the whole simulation time.

For the link state protocol, at the beginning of each simulation run, an accurate view of the network topology is installed in each mobile terminal. When the mobile terminal finds the bandwidth with its neighbor changes (due to CSI change or link breakage), it floods this change throughout the network. We aim to test the performance of the protocol and see whether it can converge or adapt to this time-varying wireless environment. Table 1 summarizes the key characteristics of the five protocols considered in our performance study. Fig. 6 shows the parameters we used for each protocol in our simulations.

3.2   Average End-to-End Delay

The first set of results is the average end-to-end delay against the mobile speed for the traffic load from 10 packets/sec to 20 packets/sec (in increments of 5),

then to 60 packets/sec, which is used for testing the performance of the protocols under an extremely heavy load. We varied the mean mobile speeds from 0 to 72 km/hr and, thus, the value of MAXSPEED was varied from 0 to 144 km/hr. As can be seen in Fig. 7, taking the CSI into consideration can greatly shorten the transmission delay from the source to destination as in BGCA and RICA. A scrutiny of the simulation traces revealed that RICA outperforms the other algorithms for the following reasons:

1. The source can update the route to the destination very frequently and usually this route is temporally the shortest one and this greatly speeds up the transmission of the data packets.

2. The periodical update of the route is adaptive to the regional changes of the CSI and this happens very frequently and, thus, in this sense, the packets may reach the destination through different routes. This means that load balancing can be achieved in RICA and each link is not overloaded for a long time, thus the buffer queue length is decreased and transmission delay is also shortened.

3. CSI checking packets sometimes make the full broadcast in search of a route unnecessary. This reduces the data queuing delay at the source because the source terminal can choose a route to destination based on CSI checking packets. In BGCA, however, route update is not so frequent as in RICA because BGCA is not as sensitive to the CSI change of the route. Consequently, BGCA cannot ensure the route to the destination is the shortest all the time. BGCA takes action (performs local query for a partial route) only when a link is in deep fading and may cause congestion.

In BGCA and RICA, the delay decreases with the increase of the speed of the mobile terminal. This is mainly because, when the mobile speed is higher, the link quality becomes worse and packets gradually get "stuck" in the buffers. However, as each packet is only allowed to be buffered for three seconds, many packets eventually get dropped and, thus, the buffer occupancy, on average, is lower than when the mobility is low. This shortens the delay of those "successful" packets (i.e., packets that

Parameter values in ABR

Hello interval	1 s
Hello message missed before link declared broken	2
Association stability threshold	500/(Average mobile speed)
Maximum relaying load on a link	2 connections
Local search timeout	0.5 s
Maximum time a data packet buffered before sent	3 s

Parameter values in AODV

Route expiration timeout	5s
Reverse route timeout	5s
Maximum times a unicast reties	3s
Broadcast Hello message	No
Mac layer link breakage detection	Yes
Maximum time a data buffer before sent	3s

Parameter values in Link State

Hello interval	1 s
Hello message missed before link declared broken	2
Routing metric	Hop count (CSI based)
Shortest routing algorithm	Dijkstra algorithm
Maximum time a data packet buffered before sent	3 s

Parameter values in BGCA

Route expiration timeout	5s
LQ reties by intermediate terminal	2
LQ timeout	0.5s
Mac layer link breakage detection	Yes
Maximum time a data buffer before sent	3s

Parameter values in RICA

CSI interval	3s
Route expiration timeout	3s
PN code detection time for possible upstream terminal	100ms
Mac layer link breakage detection	Yes
Maximum time a data buffer before sent	3s

Fig. 6. Parameters used in the protocols used in our simulations.

traverse the links when the channel quality is good). Another factor contributing to the shorter average delay is that those "slow" packets, which can reach the destinations when the mobility is low, cannot, however, reach the destinations when the mobility is high. Thus, in calculating the average, the "slow" packets are simply removed from the numerator. In ABR, however, delay increases with the mobile speed because of the local search. When the link breaks, the packets accumulate in the upstream terminal performing the local search until a partial route is found. Thus, a long buffer queue results and queuing time increases.

We also observe one interesting phenomenon in that, when mobility is low, ABR outperforms AODV. However, when mobility is high, AODV outperforms ABR in terms of end-to-end delay. This is because ABR takes the load and propagation delay of the link into consideration when selecting the route (by not choosing links with heavy load). Thus, it helps in balancing the link load and decreasing the delay. On the other hand, in AODV, the destination responds to only the first RREQ and chooses the path this RREQ has gone through, even though this route is usually

not the shortest one or some links in the route may be congested. But, as the mobility increases, the link is easier to break due to the more vigorous fading. In AODV, the source terminal performs a full broadcast in search of a new route and packets in the original broken route are discarded. As a result, a long buffer queue will not be realized.

However, in ABR, an LQ (local query [32]) is implemented to find a partial route and data packets have to wait in the terminal performing LQ. Consequently, a long buffer queue is formed, resulting in an increase in the end-to-end delay. On the other hand, the packet delivery rate of ABR is also greater than that in AODV (in AODV, a great portion of data packets is dropped due to link breakage, as observed in our experiments). Another reason is that the link in ABR is more robust than that in AODV, so a long queue is easier to form in the link with low throughput (50 kbps or 75 kbps). A long buffer queue is also formed in the link with low throughput in AODV. However, frequent link breakage subsequently eliminates these long queues. Normally, the route in ABR is longer than that in AODV (as will be seen in Section 3.5) because of their different route selection criteria.
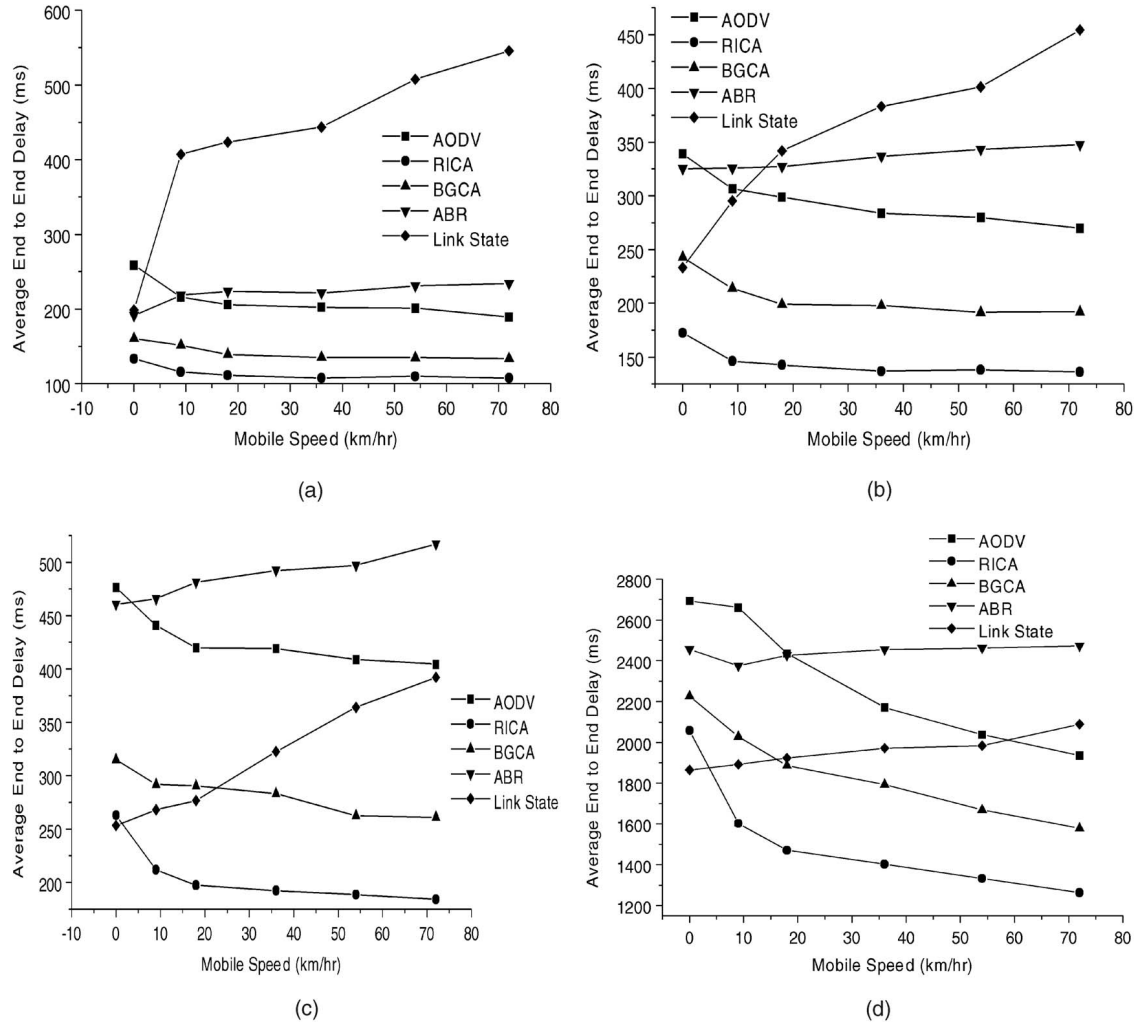
Fig. 7. Average end-to-end delays for received packets of all protocols. (a) Ten packets per second, (b) 15 packets per second, (c) 20 packets per second, and (d) 60 packets per second.

As can be seen in Fig. 7, the end-to-end delay in link state protocol increases more sharply with the mobility due to the formation of routing loop. In the link state protocol, the change of the link is broadcast throughout the network. This idea is very effective in the wire-line network where the link cost is relatively stable and the algorithm can quickly converge. However, in an ad hoc wireless network, this is not the case because the CSI or network topology changes too frequently, and each change has to be flooded as a routing packet throughout the network through the common channel. This flooding leads to an inefficient use of the common channel and the frequent collisions of the packets. Another adverse consequence is that the network status is in an inconsistent state for a prolonged period of time.

We also observe an interesting phenomenon about the link state protocol in that, as the traffic load is increased (from 10 packets/sec to 20 packets/sec), the average end-to-end delay of the packets decreases (when the mobile terminals are in motion). This is utterly different from other routing protocols. There is a plausible reason: As the mobile terminal is in motion, routing loops are formed in the network. Usually, a loop lasts for several seconds from our observations and, thus, when we increase the traffic

load, it is much easier for congestion to form in the loop because of the limited data buffer size (10 for these three scenarios). Consequently, the packets in the loop are dropped with a much higher probability and the average time a packet stays in the loop decreases. Eventually, those packets reaching the destination are from a loop-free route or a route with a loop of short lifetime. Finally, as we have expected, in all the other four algorithms, the end-to-end delay increased as we add the traffic load from 10 packets/ sec to 60 packets/sec.

3.3 Successful Percentage of Packet Delivery

From the simulation results shown in Fig. 8, we can see that taking CSI into consideration also contributes to the reliability of packet delivery. Again, in terms of successful delivery percentages, RICA outperforms the other four algorithms for the following reasons:

1. Links in RICA are with high throughput (see Section 3.5) and this ensures that data packets will not be discarded due to the long queue (e.g., buffer full).
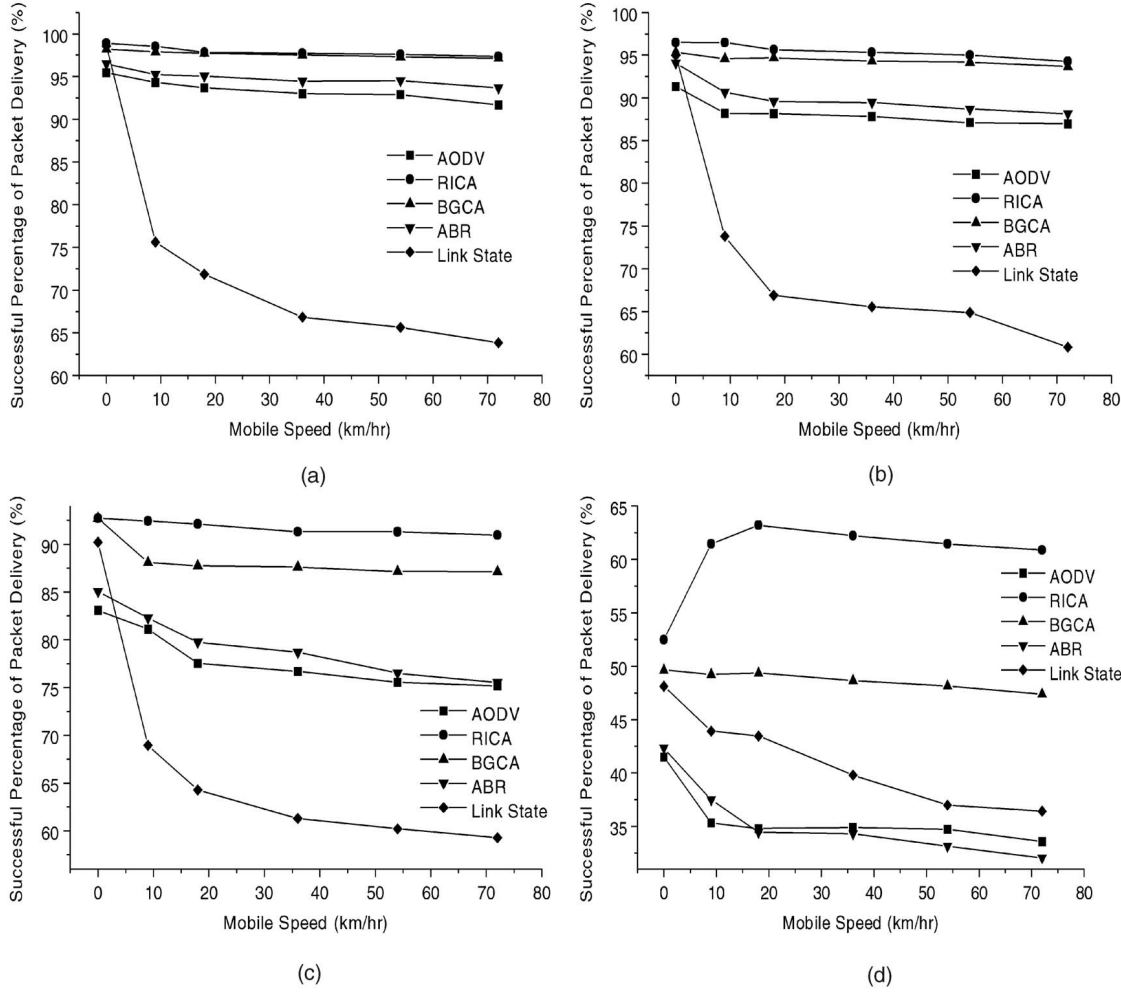
Fig. 8. Successful percentage of packet delivery of all protocols. (a) Ten packets per second, (b) 15 packets per second, (c) 20 packets per second, and (d) 60 packets per second.

2. Frequent and adaptive updates of the route can make the traffic evenly distributed in the network, thus, no link is unfairly overburdened and link congestion rarely happens.

3. Packets do not accumulate in a particular terminal because of load balancing. Thus, a long buffer queue seldom forms in a link because link breakage seldom happens.

The gain is more obvious as we increase the traffic loads. In BGCA, the update of route does not happen so often as in RICA. The route update in BGCA only happens when it is broken or in deep fading and the source has to find a new route to substitute it. Thus, the packet queues in BGCA are longer than those in RICA and when the link breaks, data loss is more serious. But, in BGCA, remedy (i.e., LQ) is taken to ensure the bandwidth requirement is satisfied. Consequently, congestion is avoided and, when, the link breaks, data loss is reduced.

ABR and AODV do not take the CSI into consideration and, therefore, their routing cannot adapt to the change of link throughput which fluctuates with time, making a long queue easier to form. In summary, there are two main causes of data loss: link congestion and not enough data buffer, and link breakage. In these two algorithms, a long

buffer queue forms very easily in the link with low throughput, especially when the traffic load is high (for example, 20 packets/sec or 60 packets/sec). We also observed the saturation of the data buffers in this circumstance many times in our experiments. As can be seen from the results of delivery rate, ABR performs better than AODV because:

● The routes in ABR are more robust than those in AODV.
● ABR takes the link load into consideration when choosing the route.
● ABR performs LQ to find a partial route at the "break point," so the probability of packets being dropped in the upstream route is reduced.

As expected, the packet delivery rate decreases with an increase of the mobility (with only one exception as described below) and traffic loads in the four on-demand routing protocols. This is because link breakage happens more often, leading to more serious congestion and long buffer queues.

In the link state routing protocol, the packet delivery rate drops more sharply with the increase of the mobile speed due to the formation of the routing loop. The higher the

mobile speed, the easier for a routing loop to form (see Section 3.5). This illustrates that link state protocol is unsuitable for a mobile wireless ad hoc network. The common channel is overused and the routing packets cannot be propagated efficiently throughout the network.

We further increase the traffic load to 60 packets/sec to test the performance of all five algorithms under this extreme condition (now, the traffic load is 245.76 kbps for a mobile terminal pair). We observe an interesting phenomenon: When the network is static, the packet delivery rate in RICA is the lowest. The optimal value is about 20 km/hr. A plausible explanation: The network is rather congested and many links are saturated (note that the maximal throughput of the link is 250 kbps) and, thus, a great portion of packets are dropped. As we further increase the mobile speed, the topology of the network changes much faster and the RICA algorithm finds more new routes. More traffic load is then shifted to these new routes. Consequently, the traffic distribution is more uniform and some links are not overloaded much (does not increase the packet delivery rate).

Under this extreme condition, link state protocol outperforms ABR and AODV protocols in packet delivery rate. This is due to the fact that the average link throughput is normally rather high in link state protocol compared with that of ABR and AODV (as illustrated below) and the links in link state routing protocol are not so congested as in ABR and AODV, although loops exist in link state protocol.

We observe that, when mobility is increased, AODV begins to outperform ABR in packet delivery rate. This is because, resulting from the extremely high load, most links in AODV and ABR are saturated (note that they cannot adapt to the CSI change of the link). Furthermore, the average length of the route in ABR is longer than that in AODV (as shown in Section 3.5). Under this extreme condition, the longer the route, the larger the number of packets being dropped.

3.4 Routing Overhead

Routing overhead is defined as the average bit rate required for sending/receiving the routing messages. The results on routing overhead are shown in Fig. 9. Taking CSI into consideration when choosing a route can improve the network performance in the sense of delay and packet delivery rate, but the cost is that it also adds more routing overhead. Using the amount of routing overhead in AODV as a baseline, BGCA and RICA generate about 1.5 and 4 times of overhead, respectively. The reason is obvious: In RICA, the destinations broadcast CSI checking packets periodically to the source so that the source can master the CSI changes timely and change the routes adaptively. In BGCA, in order to ensure the bandwidth requirement of traffic is satisfied, the intermediate terminals have to perform local search which also increases the routing overhead. As can be seen from Fig. 9, ABR generates the least amount of routing overhead because:

- The route in ABR is long-lived so the breakage of link happens not so frequently as in other routing algorithms.
- Even when the link breaks, the intermediate terminal performs local search instead of a full broadcast.

Thus, ABR can be considered rather as a bandwidth efficient algorithm. On the contrary, the amount of routing overhead in link state protocol is much higher than in other protocols. In the link state protocol, each change of the link cost is broadcast throughout the network, even though much routing information is useless. This causes a tremendous amount of routing overhead. This inefficient use of channel causes congestion and can increase the consumption of the limited battery power in each mobile terminal [30].

In all the algorithms, the routing overhead increases with the mobility because link breakage is more frequent, leading to an increase in the load of route maintenance. It is also observed that increasing the load of data traffic only has a little impact on the routing overhead because more data acknowledgments are generated.

3.5 Quality of Routes

It is also useful to compare the quality of the routes selected by different algorithms. Fig. 10a shows the average link throughput, which is defined as the total bandwidth of the links that all packets reaching destinations have passed through divided by the total number of hops that these packets have passed through. This parameter reflects the quality of the selected routes in each routing algorithm. As can be seen, the link throughputs in ABR and AODV are very close to each other and are the lowest among all protocols because these two algorithms have not taken the CSI of the link into consideration when choosing a route. In RICA and BGCA, the average link throughputs are much higher than those in ABR and AODV because the former two are adaptive to the CSI change of the link when routing packets. This is the major reason of the algorithms' ability in reducing the packet delay. We can also see that the link throughput in BGCA is a little lower than that of RICA because BGCA is a little "passive" or "conservative." That is, only when a link is in deep fading and cannot satisfy the bandwidth requirement does it take measures to find a partial route to substitute the original one. The link state routing protocol has the highest average link throughput due to the route selection criteria of the Dijkstra algorithm (when a mobile terminal needs to forward packets, it uses the shortest path algorithm to compute the next hop; normally, the link throughput between the mobile terminal and next hop is very high, for example, 250 or 150 kbps). This observation at first seems to be contradictory to the results of link state protocol, but, in fact, there is another subtle factor, as detailed below.

Fig. 10b shows the average number of hops in the route in each algorithm. The testing mobile speed is 72 km/hr for each algorithm. As can be seen, the route in link state protocol has the highest number of hops due to the formation of the routing loop. This leads to a severe deterioration of the performance even if it has the highest average link throughput. That is, even when the throughput is high, the propagation delay is so high that it offsets the gain from a higher throughput. The route in RICA has the lowest number of hops because this algorithm can continuously find the shortest route. The length of the route in ABR is longer than the other three on-demand routing protocols because ABR is inclined to select the route with
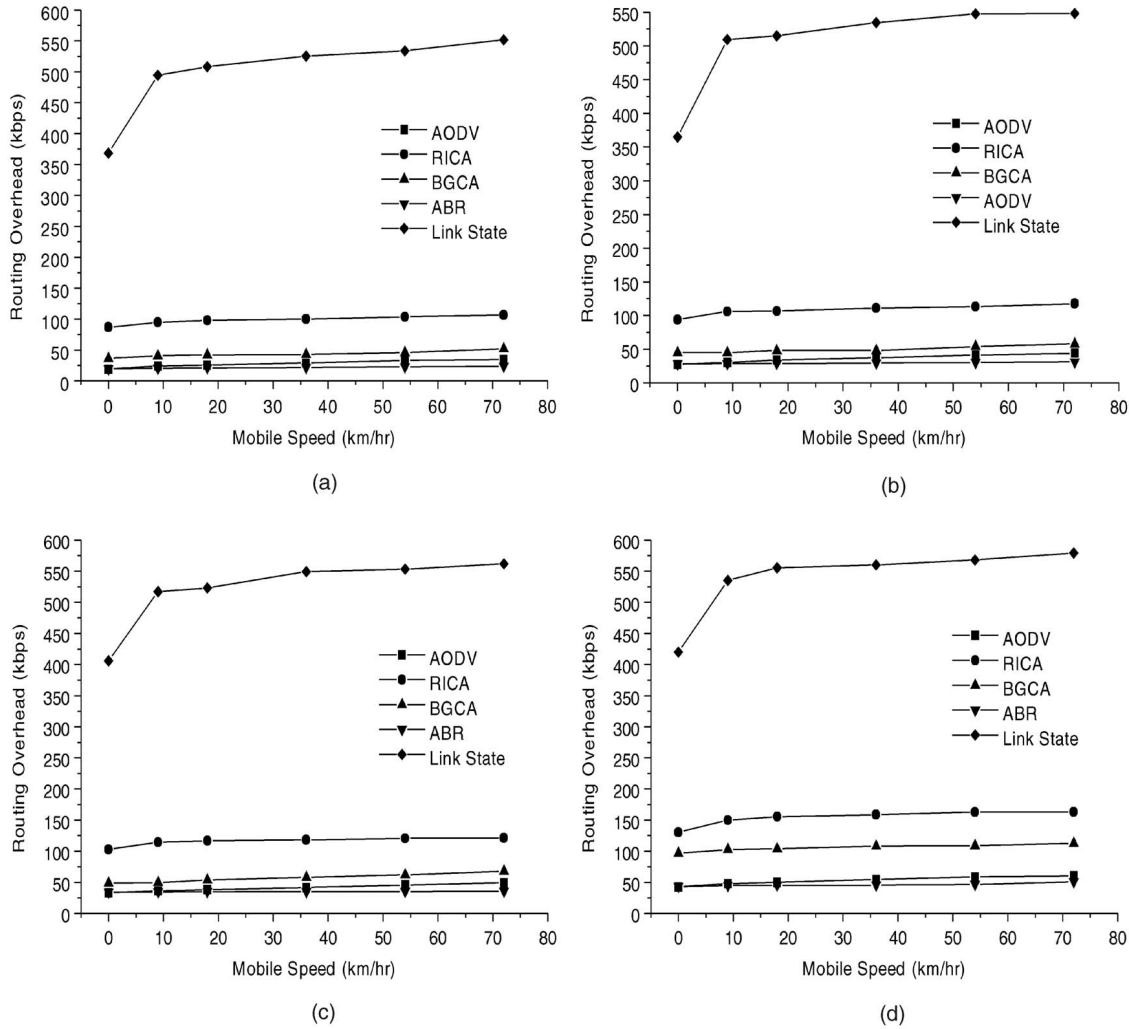
Fig. 9. Routing overhead (the average bit rate of routing messages) of all protocols. (a) Ten packets per second, (b) 15 packets per second, (c) 20 packets per second, and (d) 60 packets per second.

the highest stability and, normally, such a route has a greater number of hops.

In order to test the impact of mobile speed of mobile terminals on the route length in all protocols (in particular, the link state protocol), we also performed more experiments with various mobile speeds and the results are shown as in Fig. 10c. We can see that the number of hops in each route, in general, increases with the mobile speed. This is because, when mobile speed increases, the topology changes faster and the algorithm takes longer time to converge (it may take longer time for a loop to be eliminated). Consequently, longer end-to-end delay results. We have also tested this on the other four on-demand algorithms and found that the average number of hops in the route in these algorithms does not fluctuate greatly with the mobile speed.

Finally, as shown in Fig. 11, we also measured overall throughput, which is defined as the amount of data reaching destination terminals in every four seconds (simulation time). As can be seen, BGCA and RICA consistently outperform the other protocols in this aspect.

3.6 Overhead and Energy Efficiency

To investigate how well each protocol expends the routing overheads in terms of quality, we also perform an overhead efficiency comparison of all routing protocols. Specifically, we compute the overhead-to-throughput ratio in each protocol. The results are shown in Fig. 12a. We can see that, to achieve per bit data throughput, the amount of overhead generated in Link State is the highest (more than 0.6). On the other hand, overheads generated in other on-demand routing protocols are much less than that in Link State. In particular, in ABR, the overhead-to-throughput ratio is less than 0.1. Thus, a table-driven routing protocol is quite inefficient in terms of network overhead.

Fig. 12b shows the battery consumption variation with the elapsed time. We can see that ABR is the most conservative in power consumption, due to two factors: 1) the robustness of the routes obtained and 2) the local search mechanism. Thus, ABR seldom resorts to full broadcast in search of a route to the destination, which is very power-consuming. On the contrary, Link State employs a full broadcast throughout the network whenever a terminal finds a link change, resulting in a considerable wastage in battery energy. This is because energy depletion of Link State is the highest among all
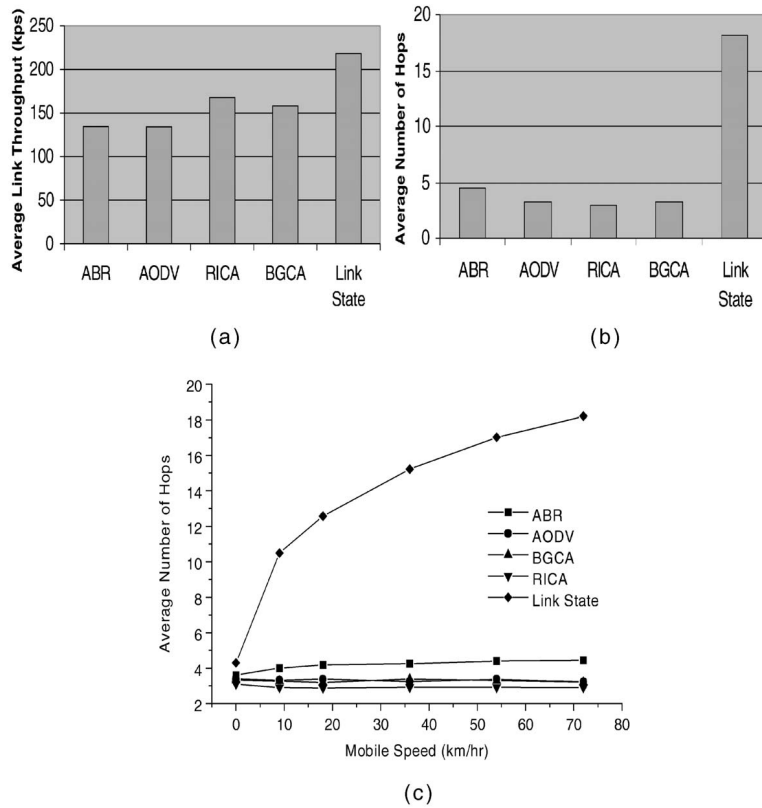
Fig. 10. Comparison of route quality of all protocols. (a) Average link throughput (at 72km/hr), (b) average number of hops (at 72km/hr), and (c) the effect of mobile speed.

protocols. Periodical packet broadcast is also adopted in RICA to make the routing from source to destination adaptive to time-varying channel quality. Thus, RICA also consumes much more power compared with other on-demand routing protocols such as AODV and ABR. However, to have a fair comparison of all routing protocols in energy efficiency, we further calculate the average power needed per Kbits of data packet reaching the destination in each protocol. The results are shown in Fig. 12c. Again, Link State is the most power-consuming protocol to achieve the same amount of network throughput, which means that frequent event-driven packet broadcast in table-driven routing protocol is inappropriate in that more power is consumed. On the other hand, ABR is again the most efficient protocol in terms of power consumption, requiring the least power per Kbits. An interesting observation is that, in RICA, the power consumption per Kbits is almost the same as that in ABR. Thus, RICA can also be considered power-efficient. Specifically, although periodical broadcast packets are used in RICA, more network throughput is achieved and, thus, the extra energy is efficiently expended.

3.7 Effect of CSI Broadcast Period

Through the periodical broadcast of CSI checking packet by destinations, RICA chooses routes with high throughputs to make the packet routing adaptive to a complicated CSI-fluctuating environment. However, we need to answer a question: How effective is a channel adaptive protocol such as RICA in view of the fact that each destination periodically broadcast CSI checking packet in such bandwidth limited

wireless environment? Could a broadcast storm occur and cause the congestion of the whole network? To avoid such a broadcast storm, the design of RICA [19] includes two measures: 1) Limit the broadcast scope of CSI checking packet by setting its TTL field as the original hop distance plus one and 2) if the CSI checking packet comes from a link in deep fading, the receiving terminal just discards this packet instead of rebroadcasts it out.

To answer the above questions more comprehensively, we have further studied the relationship between the performance of RICA and the broadcast period together with the channel characteristic. The effectiveness of a channel adaptive protocol such as RICA is constrained by two factors: 1) the "change speed" of the wireless channel and 2) the broadcast period of CSI checking packet. Here, the change speed of the wireless channel is defined as the average span time of the change of long-term fading. Obviously, the change speed of the channel has a direct impact on the routing protocol. If the channel quality changes too fast (less than 1 second, for example), no routing protocol in the network layer can react fast enough to adapt to this change in the physical layer. Thus, naturally, we want to know: What is the scope of applicability of RICA? Here, the broadcast period is also another important factor. If the period is too short, a great amount of overhead will be generated, leading to congestion of the network. If the period is too long, the terminal cannot react in a timely fashion to the change of the wireless channel. So, the question is: What is the optimal value?
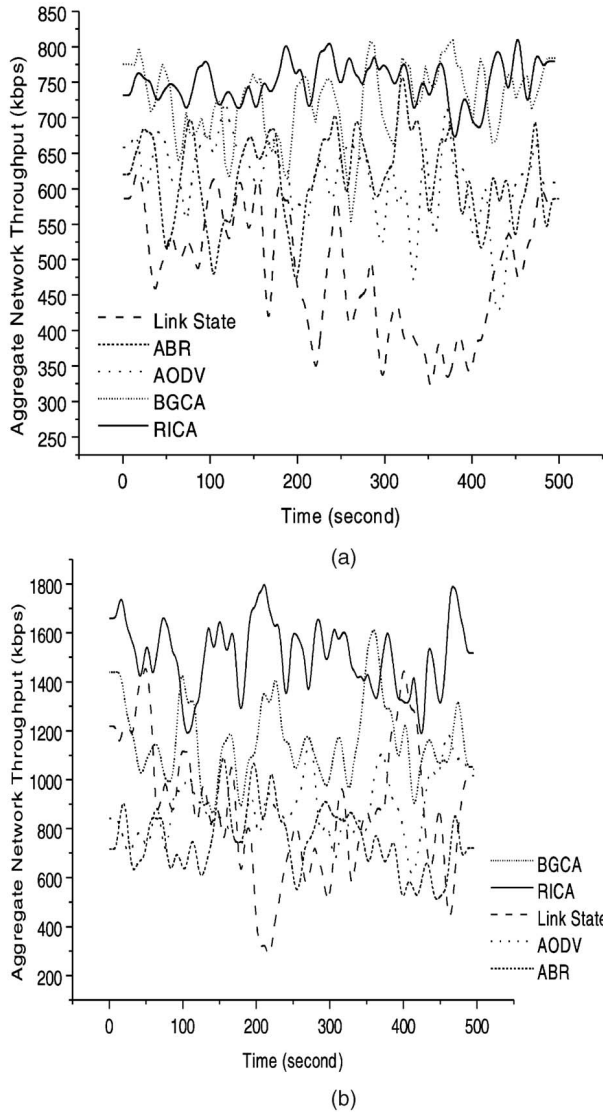
Fig. 11. Throughput variations of all the protocols over the testing time period. (a) Twenty packets per second and (b) 60 packets per second.

To obtain quantitative answers to these questions, we have done another set of extensive simulations, trying to determine the relationship between these factors and the scope of applicability of RICA. The results are shown in Table 2. We take AODV as the reference because it is the best protocol among the three other on-demand protocols as seen in the above simulation results. Here, we normalize the RICA results in that the average end-to-end delay and throughput are divided by those of AODV. For example, when the link change speed is 1 second and the CSI checking period is also 1 second, the delay achieved by RICA is 1.08 times that of AODV, indicating that the span time of a stable channel quality is so short (i.e., 1 second) that the channel adaptation performed by RICA (i.e., using CSI checking) is rather ineffective. Indeed, from the table, we can see that if the link quality changes too fast (e.g., 1 second, a situation that can happen if the background interference and/or mobility is extremely high), RICA is inapplicable as the CSI information will be out of date soon. Thus, RICA has to broadcast CSI checking packets more

frequently (every second) and this leads to congestion of the network. Only when the channel state changes relatively slowly (2 seconds or above) can RICA be applicable. The CSI broadcast period also has a direct influence on the performance, as can be seen in the table. Broadcasting too fast or too slowly is not good. We can see from the table that the optimal value is around 3 seconds. Furthermore, the data shown in Table 2 may give us some hints as to limiting the CSI broadcast period. Specifically, we can optimize the network protocol as follows: When the destination terminal communicates with its neighbor terminals, it could estimate the link change speed by using the statistical method, then link this value with its CSI broadcast period. If the change speed is fast (less than 2 seconds), the period value should be about 2 seconds. If the speed is low (3 seconds or above), then the period should be around 3 to 4 seconds.

3.8 Path Set-Up Delay

We would also like to find out how worthwhile is the extra overhead (i.e., in CSI checking) expended in the RICA protocol as compared with the AODV protocol. Fig. 13a illustrates the comparison of set-up delay of AODV and RICA. Here, set-up delay is defined as the time difference between source terminal initiating route search (constructing an RREQ) and obtaining a route (receiving an RREP). This parameter reflects how fast a protocol can set up a path to destination or achieve a route recovery. As can be seen, in the two routing protocols, the delay increases with the mobile speed. This is because, with a higher mobility, route breakage happens more frequently, leading to more route maintenance and traffic load in the common channel. We can also see that the setup delay in RICA is, as expected, much higher than that in AODV because, in RICA, the destination has to wait for a period of time in order that all the RREQs can be received. This causes delay in route setup. One point that should be mentioned here is that, in RICA, AODV, and BGCA, the delay decreases with the increase of the mobile speed. This is because we have set the packet buffer size rather small (i.e., 10) and each packet is allowed to stay in the buffer for 3 seconds only. Moreover, in the CDMA scenario considered in our study, a packet is dropped if no route to destination is available and the packet is thus not buffered. Consequently, there is a bias in the packet delay calculation: When the mobile speed is increased, packets arriving destination are normally through "good" routes (buffered time is relatively short); those packets that traversed through a link of low quality or with long buffered time are more likely discarded. The relationship of average link throughput versus mobile speed is illustrated in Fig. 13b. Average link throughput is defined as the total bandwidth of the links that all packets have passed through divided by the total number of hops that these packets have passed through. We can observe that there is a bias in the delay calculation, i.e., when the mobile speed is increased, the fraction of packets through links of high quality is also increased and those packets through "bad" routes are more prone to link congestion or link breakage.
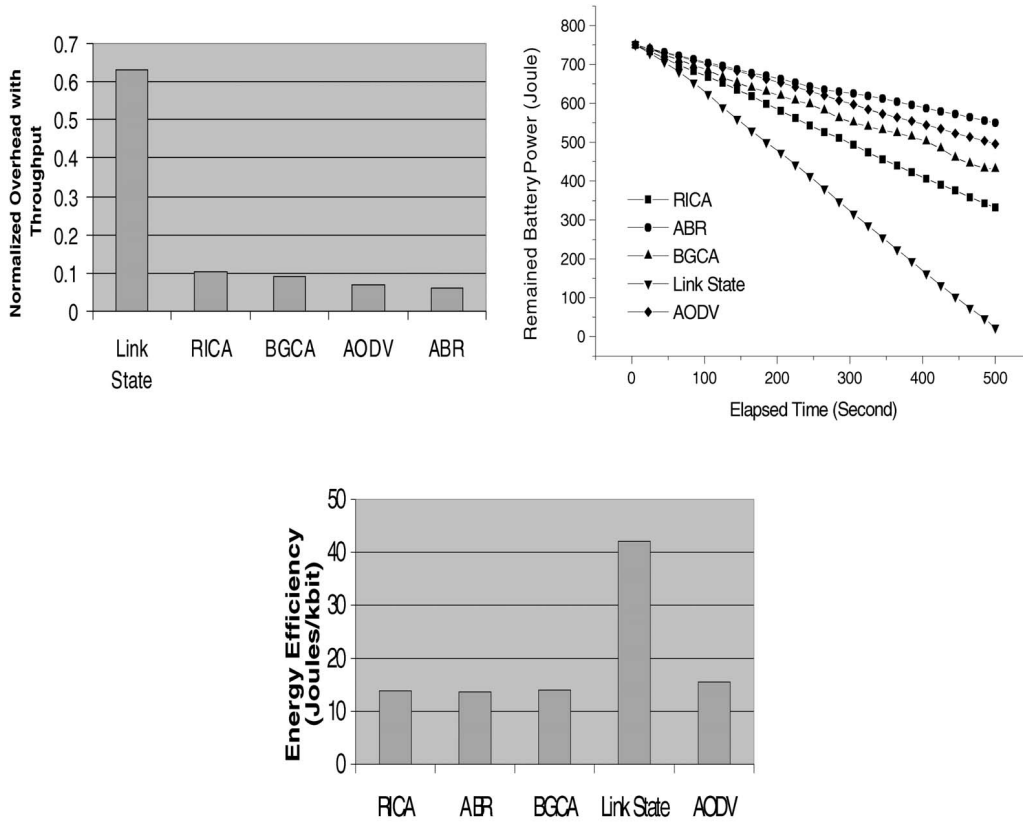
Fig. 12. Overhead and energy efficiency. (a) Overhead versus throughout, (b) energy consumption, and (c) energy efficiency.

TABLE 2
The Effect of the CSI Broadcast Period for Different Variations in Channel Quality

Normalized Performance Gain of RICA

Link Change Speed (seconds)	CSI Checking Packet Broadcast Period (seconds)									
	1		2		3		4		5	
	Delay	Throu-ghput	Delay	Throu-ghput	Delay	Throu-ghput	Delay	Throu-ghput	Delay	Throu-ghput
0.5	1.024	1.021	--	--	--	--	--	--	--	--
1	1.076	0.946	--	--	--	--	--	--	--	--
2	0.725	1.084	0.692	1.086	--	--	--	--	--	--
3	0.501	1.204	0.480	1.185	0.594	1.201	--	--	--	--
4	0.605	1.211	0.485	1.236	0.404	1.285	0.421	1.140	--	--
5	0.492	1.129	0.461	1.177	0.503	1.274	0.473	1.128	0.508	1.108

4 CONCLUDING REMARKS

In this paper, we study the behavior and performance of ad hoc routing protocols under a more realistic channel model. We compare the performance of several classical ad hoc routing protocols with that of two new routing protocols, called BGCA (Bandwidth-guaranteed Channel Adaptive) protocol and RICA (Receiver Initiated Channel Adaptive) protocol, which take into account the time-varying nature of the channel and incorporate an adaptive channel coding and modulation scheme for dynamically adjusting the throughput (the amount of error protection)

according to the channel conditions. In our extensive simulations study, we found that both the BGCA and RICA protocols outperform the well-known ABR and AODV protocols. Furthermore, the performance of RICA is slightly better than that of BGCA, indicating that changing the entire route according to channel conditions is more efficient than just conservatively and incrementally changing each link in the route. However, as the system scales up, BGCA can outperform RICA because the latter has a higher routing overhead (which is of a complexity of $O(NM)$, of which N is the number of mobile terminals and
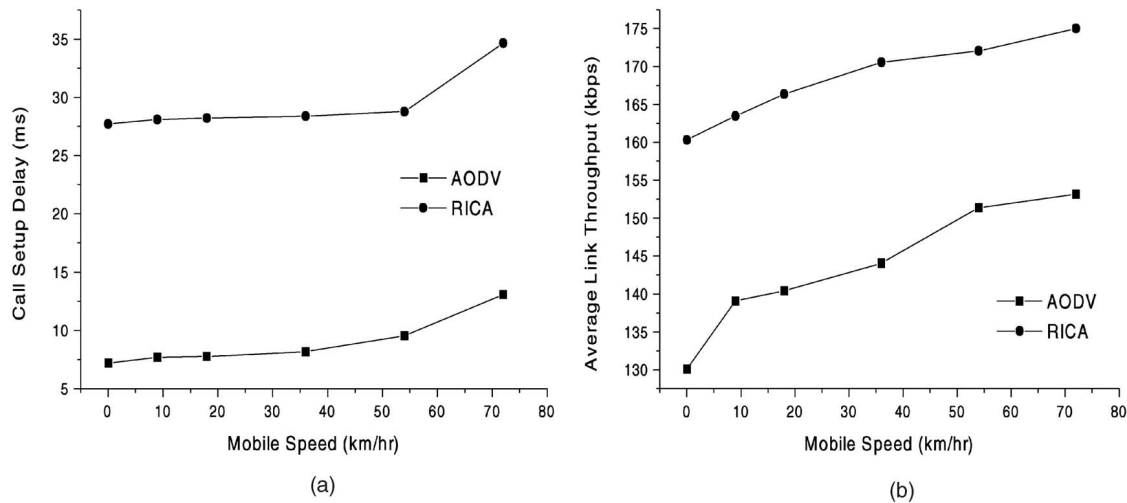
Fig. 13. Set-up delay and link quality comparison. (a) Setup delay and (b) average link throughput.

M is the number of communication terminal pairs in the network) and the common channel can be highly congested. Scalability results are not shown in this paper but can be found in [17]. On the other hand, there is possible room for further improvement for channel adaptive protocols such as RICA. If geographical information can be obtained (e.g., by using a GPS device), the destination knows the source's location (this is possible if the source sends its location to the destination), then the destination terminal can limit the propagation scope of CSI checking packets. In this manner, the destination specifies a propagation field. Only terminals in this field rebroadcast the checking packets and terminals outside the field do not broadcast the packets and, thus, the routing overhead can be greatly reduced and bandwidth can be saved. We are currently studying the performance of such a variation in the RICA protocol. Incorporating preemptive routing [9] is another major research effort under way. Another important metric we have not fully investigated in this paper is the power consumption of different protocols [20], [35].
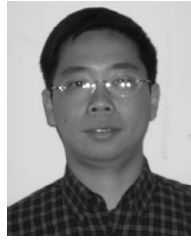
ACKNOWLEDGMENTS

REFERENCES

[1] M.-S. Alouini, X. Tang, and A.J. Goldsmith, "An Adaptive Modulation Scheme for Simultaneous Voice and Data Transmission over Fading Channels," *IEEE J. Selected Areas in Comm.,* vol. 17, no. 5, pp. 837-850, May 1999.
[2] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proc. MOBICOM '98,* pp. 85-97, July 1998.
[3] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms,* second ed. MIT Press, 2001.
[4] M.S. Corson, J.P. Macker, and G.H. Cirincione, "Internet-Based Mobile Ad Hoc Networking," *IEEE Internet Computing,* pp. 63-70, July/Aug. 1999.
[5] D.S.J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing," *Proc. ACM MOBICOM 2003,* pp. 134-146, Sept. 2003.
[6] R. Dube, C.D. Rais, and S.K. Tripathi, "Improving NFS Performance over Wireless Links," *IEEE Trans. Computers,* vol. 46, no. 3, pp. 290-298, Mar. 1997.
[7] R. Dube, K.-Y. Wang, C.D. Rais, and S.K. Tripathi, "Signal Stability Based Adaptive Routing (SSA) for Ad Hoc Mobile Networks," *IEEE Personal Comm.,* vol. 4, no. 1, pp. 36-45, Feb. 1997.
[8] J.J. Garcia-Luna-Aceves and S. Murthy, "A Path-Finding Algorithm for Loop-Free Routing," *IEEE/ACM Trans. Networking,* vol. 5, no. 1, pp. 148-160, Feb. 1997.
[9] T. Goff, N. Abu-Ghazaleh, D. Phatak, and R. Kahvecioglu, "Preemptive Routing in Ad Hoc Networks," *Proc. MOBICOM '01,* pp. 43-52, July 2001.
[10] H.-Y. Hsieh and R. Sivakumar, "Routing: On Using the Ad Hoc Network Model in Cellular Packet Data Networks," *Proc. ACM MOBIHOC '02,* pp. 35-47, June 2002.
[11] C. Huitema, *Routing in the Internet,* second ed. Prentice Hall, 2000.
[12] A.D. Joseph, J.A. Tauber, and M.F. Kaashoek, "Mobile Computing with the Rover Toolkit," *IEEE Trans. Computers,* vol. 46, no. 3, pp. 337-352, Mar. 1997.
[13] K.I. Kim, *Handbook of CDMA System Design, Engineering, and Optimization.* Prentice Hall, 2000.
[14] Y.-K. Kwok and V.K.-N. Lau, "A Novel Channel-Adaptive Uplink Access Control Protocol for Nomadic Computing," *IEEE Trans. Parallel and Distributed Systems,* vol. 13, no. 11, pp. 1150-1165, Nov. 2002.
[15] V.K.N. Lau, "Performance Analysis of Variable Rate Symbol-by-Symbol Adaptive Bit-Interleaved Coded Modulation for Rayleigh Fading Channels," *IEEE Trans. Vehicular Technology,* vol. 51, no. 3, pp. 537-550, May 2002.
[16] S.-J. Lee, M. Gerla, and C.-K. Toh, "A Simulation Study of Table-driven and On-Demand Routing Protocols for Mobile Ad Hoc Networks," *IEEE Network,* vol. 13, no. 4, pp. 48-54, July-Aug. 1999.
[17] X.-H. Lin, "Channel Adaptive Techniques for Resources Management in Wireless Ad Hoc Networks," PhD thesis, The Univ. of Hong Kong, Aug. 2003.
[18] X.-H. Lin, Y.-K. Kwok, and V.K.N. Lau, "BGCA: Bandwidth Guarded Channel Adaptive Routing for Ad Hoc Networks," *Proc. Third IEEE Comm. and Networking Conf. (WCNC '02),* vol. 1, pp. 433-439, Mar. 2002.
[19] X.-H. Lin, Y.-K. Kwok, and V.K.N. Lau, "RICA: A Receiver-Initiated Approach for Channel-Adaptive On-Demand Routing in Ad Hoc Mobile Computing Networks," *Proc. 22nd IEEE Int'l Conf. Distributed Computing Systems (ICDCS '02),* pp. 84-91, July 2002.

[20] X.-H. Lin, Y.-K. Kwok, and V.K.N. Lau, "Power Control for IEEE 802.11 Ad Hoc Networks: Issues and a New Algorithm," *Proc. 2003 Int'l Conf. Parallel Processing,* Oct. 2003.

[21] Q. Lu and M. Satyanarayanan, "Resource Conservation in a Mobile Transaction System," *IEEE Trans. Computers,* vol. 46, no. 3, pp. 299-311, Mar. 1997.

[22] D.A. Maltz, "On-Demand Routing in Multi-Hop Wireless Mobile Ad Hoc Networks," PhD thesis, Carnegie Mellon Univ., 1998.

[23] Network Simulator (NS-2), http://www.isi.edu/nsnam/ns/, 2003.

[24] K. Pahlavan and P. Krishnamurthy, *Principles of Wireless Networks: A Unified Approach.* Prentice Hall, 2003.

[25] J.D. Parsons, *The Mobile Radio Propagation Channel,* second ed. Wiley, 2000.

[26] *Ad Hoc Networking,* C.E. Perkins, ed. Addison-Wesley, 2000.

[27] C.E. Perkins and E.M. Royer, "Ad-Hoc On-Demand Distance Vecotr Routing, Mobile Computing Systems and Applications," *Proc. IEEE Workshop Mobile Computing Systems & Applications (WMCSA '99),* pp. 90-100, 1999.

[28] E.M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," *IEEE Personal Comm.,* vol. 6, no. 2, pp. 46-55, Apr. 1999.

[29] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Media Access Control for Ad Hoc Networks: Opportunistic Media Access for Multirate Ad Hoc Networks," *Proc. ACM MOBICOM '02,* pp. 24-35, Sept. 2002.

[30] A.P. Sista, O. Wolfson, and Y. Huang, "Minimization of Communication Cost through Caching in Mobile Environments," *IEEE Trans. Parallel and Distributed Systems,* vol. 9, no. 4, pp. 378-390, Apr. 1998.

[31] C.-K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems.* Prentice Hall, 2002.

[32] C.-K. Toh, "A Novel Distributed Routing Protocol to Support Ad-Hoc Mobile Computing," *Proc. 1996 IEEE 15th Ann. Int'l Phoenix Conf. Computers and Comm.,* pp. 480-486, 1996.

[33] C.-K. Toh, M. Delwar, and D. Allen, "Evaluating the Communication Performance of an Ad Hoc Wireless Network," *IEEE Trans. Wireless Comm.,* vol. 1, no. 3, pp. 402-414, July 2002.

[34] J. Yoon, M. Liu, and B. Noble, "Random Waypoint Considered Harmful," *Proc. INFOCOM,* 2003.

[35] M. Zorzi and R.R. Rao, "Error Control and Energy Consumption in Comm. for Nomadic Computing," *IEEE Trans. Computers,* vol. 46, no. 3, pp. 279-289, Mar. 1997.

Xiao-Hui Lin received the BS and MS degrees in electronics and information science from Lanzhou University in 1997 and 2000, respectively. He received the PhD degree in electrical and electronic engineering from The University of Hong Kong in 2003. He is now an assistant professor in the College of Electrical Engineering, South China University of Technology, Guangzhou, China. His research interests include mobile computing, ad hoc wireless networks, and multimedia communications.

Yu-Kwong Kwok received the BSc degree in computer engineering from The University of Hong Kong in 1991, and the MPhil and PhD degrees in computer science from the Hong Kong University of Science and Technology (HKUST) in 1994 and 1997, respectively. He is an associate professor in the Department of Electrical and Electronic Engineering at The University of Hong Kong. Before joining The University of Hong Kong, he was a visiting scholar in the parallel processing laboratory at the School of Electrical and Computer Engineering at Purdue University. His research interests include mobile computing, wireless networking, and distributed computing systems. He is a senior member of the IEEE. He is also a member of the ACM, the IEEE Computer Society, and the IEEE Communications Society.

Vincent K.N. Lau received the BEng (Distinction 1st Honors) in electrical engineering in 1992 from The University of Hong Kong. He joined HK Telecom after graduation and worked there for three years as a system engineer, responsible for transmission systems design. He was awarded the Sir Edward Youde Memorial Fellowship and the Croucher Foundation in 1995 and received the PhD degree in 1997 from the University of Cambridge. He joined the Lucent Technologies-Bell Labs in the US as a member of the technical staff and was engaged in the algorithm design, standardization, and prototype development of CDMA2000 systems. He joined The University of Hong Kong in 1999 as an assistant professor and was appointed the codirector of the information engineering programme as well as the codirector of the 3G Technology Center. In July 2001, he left the university and returned to the the Wireless Advanced Technology Lab at Lucent Technologies. His research interests include digital transceiver design, adaptive modulation and channel coding, CDMA power control, soft handoff and CREST factor control algorithms, jointly adaptive multiple access protocols, as well as short-range wireless ad-hoc networking. He is currently working on BLAST-MIMO systems, iterative decoding and UMTS call processing protocol stack design. He is a senior member of the IEEE and the IEEE Communications Society.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.